(19) Japanese Patent Office (JP)
(12) Public Patent Disclosure Bulletin (A)
(11) Public Patent Disclosure Bulletin No.: H08 - 202626
(43) Disclosure Bulletin Date: August 9, 1996

(51) Int. Cl.		Identification Code	JPO File No.	FI	Tech. Indic.
G 06 F	12/16	3 1 0 A	7623 - 5B		
G 11 C	16/06				
	29/00	3 0 1 A			
				G 11 C 17/00 309 F	

Request for Examination: NO, Claims: 2 (Total Pages: 7)

(21) Patent Application No.: H07-013697

(22) Patent Application Date: January 31, 1995

(71) Applicant: 000000170
Isuzu Motors Ltd.
6-26-1 Minamiooi,
Shinagawaku, Tokyo

(72) Inventor: Toshibumi Koshizawa
8 Tsuchidana, Fujisawa,
Kanagawa Prefecture
Isuzu Central Research (in-house)

(74) Representative: Shuuji Moizumi, Attorney-at-Law

(54) Title of the Invention: Memory Controller

(57) Abstract
[Purpose] In a memory controller that writes out data to an EEPROM, data that needs to be rewritten frequently over a long period is stored correctly.
[Construction] The data [in the CPU] is compared with the data already written to the same address of the EEPROM and if both are the same the processing stops, or the frequency at which data is written out to the same address of the EEPROM is counted and stored in the EEPROM and, when the count value reaches the reload limit frequency, the address of the data is modified and written out.

[Figure]
Embodiment (1) of the Invention

3 - Input Data
4 - Common Bus

Claims

Claim 1. A memory controller for writing out data to an EEPROM wherein said data is compared with data already written out to the same address of said EEPROM and, if both are the same, an update is blocked.

Claim 2. A memory controller for writing out data to an EEPROM wherein the frequency at which said data is written out to the same address of said EEPROM is counted and stored in said EEPROM and, when said count value reaches the write-out limit frequency, the address of said data is modified and written out.

Detailed Description of the Invention

[0001]

Industrial Field of Application

This invention concerns a memory controller and, particularly, a memory controller used in electronic devices installed in vehicles.

[0002]

Prior Art

In electronic devices for vehicles, learning control parameters, settings, etc. are checked in order to optimize their control, these are stored in a memory for reference in checking and servicing a failure code corresponding to the details of the system failure.

[0003]

With such learning data and failure status codes, data needs to be stored, of course, when the vehicular electronic device is in operation and while the device is resting when the main power source is OFF.

[0004]

Because of this, conventionally, in order to realize the storing of learning data and failure status codes, at first a method utilizing a memory backup power source was adopted.

[0005]

In this method a RAM is installed in an electronic device and the main power source of the electronic device source provides a power source to this RAM from a separate system power source (backup power source). Even when the electronic device is resting (power source OFF) RAM data is stored by the supply to the backup power source.

[0006]

However, the method utilizing such a memory backup power source has the following pitfalls.

[0007]

1. It is necessary to extend the wiring of the main power source to the backup power source in a separate system.
2. As wiring for a backup power source normally brings about an application of power source voltage, in consideration of prevention of electrical accidents this wiring should not be done.
3. Since the backup power supply is in direct contact with the vehicular battery, if the battery is disconnected when there is inspection and servicing unrelated to inspection and servicing of electronic devices, the data in memory is erased.
4. Since data in memory is erased when the body of an electronic device, even if restorable, is removed from a vehicle due to a failure, it is necessary to write down in a memo or whatever the contents of memory before removing the electronic device when taking the trouble to reproduce or confirm a failure.

[0008]

Because of these things, in recent years a method using an EEPROM instead of a memory backup power source has become common.

[0009]

That is, by using an EEPROM known as an electrically erasable/writable non-volatile memory device, the data that needs to be stored is written out to this EEPROM; this method of performing data storage has been used a lot.

[0010]
Problems the Invention Is To Solve
However, in the method of using the EEPROM like this there is the following problem.
[0011]
With the internal construction of the EEPROM, because the memory element gradually deteriorates due to erasing/writing-out data, the frequency of erase/write-out operations of data at the same address of the EEPROM is limited, so it is not suitable when it is necessary to write out data frequently.
[0012]
This point is a big problem in electronic devices for vehicles which require a comparatively long operational period and usable lifetime.
[0013]
Subsequently, this invention has the purpose whereby, in a memory controller that writes out data to an EEPROM, data that needs to be rewritten frequently over a long period is stored correctly.
[0014]
Means of Solving the Problems
(1) In order to achieve the above aim, in the memory controller relating to this invention, data is compared with data already written out to the same address of an EEPROM and, if both are the same, an update is blocked.
[0015]
(2) And, in the invention, the frequency at which said data is written out to the same address of the EEPROM is counted and stored in said EEPROM and, when said count value reaches the write-out limit frequency, the address of said data is modified and written out.

[0016]
Operation
(1) In electronic devices, internal data calculations are performed frequently and, when changes of data are few and the data calculated each time is nearly uniform, the data already written to EEPROM after data calculation is read once. Comparing this with the calculated data, only if both are different, the contents of the EEPROM already written are erased and the new data is written out.
[0017]
(2) As state above, when data is calculated frequently and there are many changes of data, several EEPROM addresses are provided and, when writing out data regarding the same address, the frequency is counted and this count is stored in the EEPROM.
[0018]
Also, when this count value exceeds the erase/write-out frequency stored in the EEPROM, by setting the address of the EEPROM anew, the same address cannot be used above a prescribed frequency. In this way the longevity of the EEPROM is provided for.
[0019]
Embodiment
Figure 1 shows an embodiment of the memory controller [2] relating to this invention and in the figure: [1] shows the CPU of the memory controller, [2] shows the EEPROM memory element, [3] shows the interface circuit for receiving data input, [4] shows the common bus providing data from the interface circuit [3] to the CPU [1] and the EEPROM [2].
[0020]
Moreover, Figure 2 show the memory map of the EEPROM [2] for data read out according to the invention and here the data that is to be read out is devised as a 1-byte unit. Thus, as shown in fig. 2, only the address [ADD] for data write is necessary for the EEPROM [2] memory.
[0021]
Figure 3 shows a flow chart of the data write-in operation handled by the CPU [1] shown in Fig. 1 and this flow chart is operative when data is being written in.

[0022]
First, the CPU [1] calculates the data (A) to be written in (step S1).
[0023]
Then, the contents (B) of the address [ADD] already stored in the EEPROM [2] are read out (step S2).
[0024]
Then, there is a comparison (step S3) of the data calculated in step S1 with the memory contents of the EEPROM [2] read out in step S2.
[0025]
As a result, if A = B a write out of data is not necessary and the processing ends but, if A ≠ B the data (A) calculated in step S1 is written out to the EEPROM [2] and the contents of the address [ADD] are erased (step S4).
[0026]
So, the data (A) is written to the address [ADD] (step S5) and the processing is completed.
[0027]
By the process of the above steps S1 ~ S5, generally the same value as the most recent data (A) can be stored in the EEPROM [2] but, in this case there is a modification of data and, since an erase/write-out is performed only when necessary, there is no adverse effect on the lifetime of the EEPROM.
[0028]
Figure 4 shows a flow chart of when data is read in by the CPU [1] by the EEPROM [2] and the data read reads the contents directly from the address [ADD] similar to the data read out (as in step S2) of an ordinary RAM (step S10).
[0029]
Figure 5 shows the embodiment of the memory controller (2) relating to the invention and, adding to the embodiment of the invention shown in Fig. 1, it differs in that it has a 2-byte register [R1] and a 1-byte register [R2] in the CPU [1]. A ROM [10] storing a prescribed frequency is included in the CPU [1] as well.
[0030]
Figure 6 shows the memory map of the EEPROM [2] shown in Fig. 5; the offset value (point data) for showing the memory address and the address of the frequency count of the memory address' erase/write-out in the address [ADD].
[0031]
Further, as shown in the figure, the memory for the data to be read out and the memory for the erase/write- out frequency count are allocated to the array X successively.
[0032]
Here the data to be written out is 1-byte, and 2-bytes is allocated for the erase/write-out frequency count as the maximum value of the erase/write-out frequency is 65,535. Thus, the write-out frequency count value is written to X(0), X(1) and X(3), X(4), ... and the data is written to addresses X(2), X(5), ...
[0033]
In Figure 7, first the data (A) to be written out is calculated (step S11).
[0034]
Then, the erase/write-out frequency count value is read and loaded into the register [R1] (step S12). In this case, as shown in the Fig. 6, as the count value is stored in all 2-bytes of the memory address where the data is written, the count value is stored in the addresses, X(0 + ADD) and X(1 + ADD).
[0036]
Next, the count value of the register [R1] and the set value are compared (step S13).
[0037]
This set value is previously stored in the ROM [10] built into the CPU [1] and, as mentioned above, it is set according to the erase/write-out frequencies guaranteed in the EEPROM [2]; in this embodiment for 2-byte data this could be set to a maximum of 65,535 times. This is not to say that when this frequency is guaranteed above 65, 535 times, it could not be modified to set 3-bytes of memory and so forth for the count.

[0038]
As a result of comparison in step S13, when the contents of the register [R1] is less than the set value, since the count value and data can be written out to memory at the same address as the erase/write-out of the count and data together has been done up to now, "1" is added to the present erase/write-out frequency and this result remains in the register [R1] (step S14).
[0039]
Then, to write in the next count value, the contents of addresses, X(0 + ADD) and X(1 + ADD), are erased (step S15).
[0040]
Next, the contents of the register [R1] are written to the addresses, X(0 + ADD) and X(1 + ADD), (step S16), to write out the data (A) the contents of the memory address for data storage, X(2 + ADD), are erased (step S17), the data (A) is written to the address, X(2 + ADD), (step S18), and the process is completed.
[0041]
On the other hand, in step S13, when it is determined that the contents of the register [R1] reach the set value, the operations of steps S19 ~ S23 are carried out to modify the addresses of the memory for data and the memory for the count.
[0042]
First, in step S19, the offset of the address [ADD], the address point, is transferred to the register [R2].
[0043]
Then, to write out the offset in step S11, the offset value of the address [ADD] is erased, "3" is added to the address [ADD] presently transferred to the register [R2] and the result of this remains in the register [R2] (step S21).
[0044]
In step S22 the value of that register [R2] is written to the address [ADD] as a new address offset value; further, as it is the first data write-out to a new address, "1" is written to the register [R1] as the erase/write-out frequency count (step S23).
[0045]
After this, jumping to step S15, the same operations as above (steps S15 ~ S18) are carried out and the process stops.
[0046]
Thus, by executing the operations of steps S19 ~ S23, just "3" is added to the current point value (offset of address [ADD]). As the count value address and data address on the memory map of the EEPROM [2] shown in Fig. 6 are respectively modified by "3", a new address is set.
[0047]
Further, since the address point value (contents of address [ADD]) and erase/write-out frequency count value (address X(0 + ADD), X(1 + ADD)) together with the data value (address X(2 + ADD)) are stored in the EEPROM [2], even when the main power source of the electronic device is OFF, the erase/write-out frequency is counted accurately in the EEPROM [2] and the process ends without the disappearance of the stored data.
[0048]
Figure 8 shows a flow chart of the read-out of the data value written by the operation shown in Fig. 7 and in reading the memory contents of the address, X(2 + ADD), the stored data value can be read (step S30).
[0049]
In the memory map shown in Fig. 6 and the flow chart shown in Fig. 7, the address on the memory map of the address offset of the address [ADD] is fixed at the first place but, as it is set in the EEPROM, there is a limit on the erase/write-out frequency similarly to the data and it can be modified in the same way as the data address is modified.

[0050]

Effect of the Invention

With the memory controller of the invention, as described above, because of a constitution whereby when, comparing data with data already written to the same address of an EEPROM, both are the same, processing stops and when, after storing in the EEPROM the data of the frequency count of erase/write-outs to the same address of the EEPROM, said count value reaches the write-out frequency limit, it is written out to a modified data address, a backup power source is not necessary and, additionally, data that needs to be written out frequently can be stored accurately for the long term. In particular, learning data and failure status codes can be stored without a backup power source in electronic devices for commercially produced vehicles.

A Brief Description of the Figures

Figure 1 is a block diagram showing the embodiment of the memory controller (1) relating to this invention.

Figure 2 is a memory map of the EEPROM used in the memory controller (2) relating to the invention.

Figure 3 is a flow chart of the data write operation carried out by the CPU in the memory controller (1) relating to the invention.

Figure 4 is a flow chart of the data read operation carried out by the CPU in the memory controller (1) relating to the invention.

Figure 5 is a block diagram showing an embodiment of the memory controller (2) relating to the invention.

Figure 6 is memory map of the EEPROM used in the memory controller (2) relating to the invention.

Figure 7 is a flow chart of the data write operation carried out by the CPU in the memory controller (2) relating to the invention.

Figure 8 is a flow chart of the data read operation carried out by the CPU in the memory controller (2) relating to the invention.

Explanation of Reference Numbers

1 CPU
2 EEPROM
10 ROM
R1, R2 Registers

In the figures, the same reference numbers show the same or related parts.

[Figure 1] Embodiment (1) of the Invention
3 Input Data
4 Common Bus

[Figure 2] EEPROM Memory Map
Address [ADD] 1-byte

[Figure 3] Flow Chart of Data Write Operation
S1 Calculate data (A) to be written
S2 Read contents (B) in memory at address [ADD]
S3 Compare A and B
S4 Erase contents of address [ADD]
S5 Write out data (A) to be written to address [ADD]

[Figure 4] Flow Chart of Data Read Operation
S10 Read the contents of memory at address [ADD]

[Figure 5] Embodiment (2) of the Invention
1 R1 register
R2 register
3 Input data

4 Common Bus

[Figure 6] EEPROM Memory Map
Address [ADD] Address offset value (1-byte)
Write-out frequency count (2-byte)
Data (1-byte)
Write-out frequency count (2-byte)
Data (1-byte)

[Figure 8] Flow Chart of Data Read Operation
S30 Read the memory contents of the address, X(2 + ADD)

[Figure 7] Flow Chart of Data Write Operation
S11 Calculate data (A) to be written
S12 Read 1write-out frequency count to register R
S13 Contents of register R1

[left branch] < set value
S15 Erase contents of addresses, X(0 + ADD), X(1 + ADD)
S16 Write contents of register R! to addresses, X(0 + ADD), X(1 + ADD)
S17 Erase contents of address, (2 + ADD)
S18 Write data (A) to be written to address, (2 + ADD)

[right branch] = set value
S19 Contents of address [ADD] → R2
S20 Erase contents of address [ADD]
S22 Write contents of R2 to address [ADD]